

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-301787

(43) 公開日 平成10年(1998)11月13日

(51) Int.Cl.⁶

G 0 6 F 9/445

9/06

識別記号

5 3 0

F I

G 0 6 F 9/06

4 2 0 J

5 3 0 W

審査請求 未請求 請求項の数10 O L (全 9 頁)

(21) 出願番号 特願平9-110635
 (22) 出願日 平成9年(1997)4月28日

(71) 出願人 000005108
 株式会社日立製作所
 東京都千代田区神田駿河台四丁目6番地
 (71) 出願人 000233136
 株式会社日立画像情報システム
 神奈川県横浜市戸塚区吉田町292番地
 (71) 出願人 000153476
 株式会社日立マイクロソフトウェアシステムズ
 神奈川県横浜市戸塚区吉田町292番地
 (74) 代理人 弁理士 小川 勝男

最終頁に続く

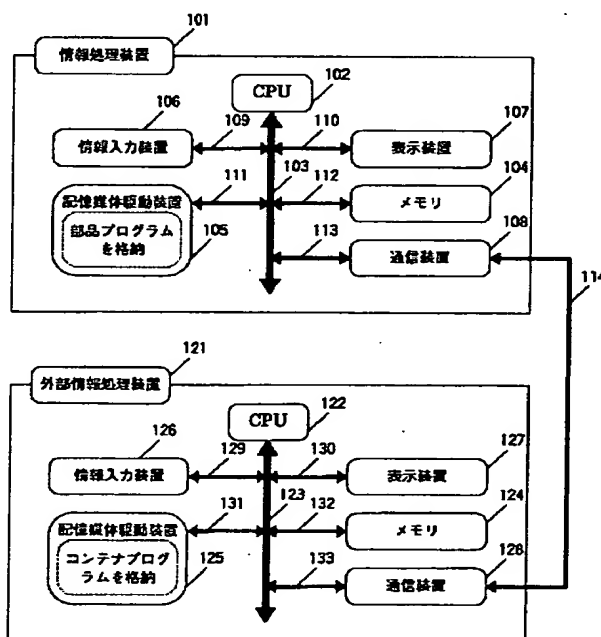
(54) 【発明の名称】 情報処理装置

(57) 【要約】

【課題】 ネットワーク経由でソフトウェアを高速に入
 手、実行する。

【解決手段】 演算処理等を行なう中央演算装置と、演算データ、演算結果や制御プログラムを記憶するメモリと、表示データの再生処理を行なう表示装置と、情報記憶媒体及びその情報記憶媒体の駆動装置と、外部からの特定の入力を行なうための情報入力装置と、外部の情報処理装置との通信を行なう通信装置を備えるアプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、情報記憶媒体に記憶されたソフトウェア部品の結合情報を上記通信装置経由で外部の情報処理装置から入手し、実行可能形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行することにある。

図 1



1

【特許請求の範囲】

【請求項1】演算処理等を行なう中央演算装置と、演算データ、演算結果や制御プログラムを記憶するメモリと、表示データの再生処理を行なう表示装置と、情報記憶媒体及びその情報記憶媒体の駆動装置と、外部からの特定の入力を行なうための情報入力装置と、外部の情報処理装置との通信を行なう通信装置を備える情報処理装置において、

アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、アプリケーションソフトウェアの実行の際に、アプリケーションソフトウェアの機能を実現するのに必要な情報を、上記通信装置経由で外部の情報処理装置から入手し、実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行することを特徴とする情報処理装置。

【請求項2】請求項1記載の情報処理装置であって、アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、アプリケーションソフトウェアの実行の際に、情報記憶媒体に記憶された上記ソフトウェア部品のうち、機能を実現するのに必要となるソフトウェア部品を選択して組み上げ実行するのに必要な情報を、上記通信装置経由で外部の情報処理装置から入手し、実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行することを特徴とする情報処理装置。

【請求項3】請求項1記載の情報処理装置であって、アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、アプリケーションソフトウェアの実行の際に、情報記憶媒体に記憶された上記ソフトウェア部品のうち、機能を実現するのに必要となるソフトウェア部品を選択して組み上げ実行するのに必要な情報を、上記通信装置経由で外部の情報処理装置から入手し、

かつ、アプリケーションソフトウェアの実行に必要なソフトウェア部品が情報記憶媒体の中に存在しない場合に、そのソフトウェア部品を上記通信装置経由で外部の情報処理装置から入手し、実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行することを特徴とする情報処理装置。

【請求項4】請求項1、2、又は3記載の情報処理装置であって、かつ、情報記憶媒体の駆動装置が扱う情報記憶媒体が読み出し、書き込み可能な媒体であって、アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、

アプリケーションソフトウェアの実行の際に、情報記憶媒体に記憶された上記ソフトウェア部品のうち、機能を実現するのに必要となるソフトウェア部品を選択して組み上げ実行するのに必要な情報を、上記通信装置経由で

2

外部の情報処理装置から入手し、

実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行すると共に、上記実行可能な形式に組み上げられたソフトウェアを、上記読み出し、書き込み可能な記憶媒体に記憶し、次回実行時には、上記記憶媒体に保存されたアプリケーションソフトウェアを呼び出すことを特徴とする情報処理装置。

【請求項5】請求項3記載の情報処理装置であって、かつ、情報記憶媒体の駆動装置が扱う情報記憶媒体が読み出し、書き込み可能な媒体であって、アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、アプリケーションソフトウェアの実行の際に、情報記憶媒体に記憶された上記ソフトウェア部品のうち、

機能を実現するのに必要となるソフトウェア部品を選択して組み上げ実行するのに必要な情報を、上記通信装置経由で外部の情報処理装置から入手し、かつ、アプリケーションソフトウェアの実行に必要なソフトウェア部品が情報記憶媒体の中に存在しない場合に、そのソフトウェア部品を上記通信装置経由で外部の情報処理装置から入手し、

実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行すると共に、上記の外部情報処理装置から入手したソフトウェア部品を、上記読み出し、書き込み可能な記憶媒体に記憶し、次回実行時には、上記記憶媒体に保存されたソフトウェア部品を呼び出すことを特徴とする情報処理装置。

【請求項6】請求項3記載の情報処理装置であって、かつ、情報記憶媒体の駆動装置が扱う情報記憶媒体が読み出し、書き込み可能な媒体であって、アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、アプリケーションソフトウェアの実行の際に、情報記憶媒体に記憶された上記ソフトウェア部品のうち、機能を実現するのに必要となるソフトウェア部品を選択して組み上げ実行するのに必要な情報を、上記通信装置経由で外部の情報処理装置から入手し、

かつ、アプリケーションソフトウェアの実行に必要なソフトウェア部品が情報記憶媒体の中に存在しない場合に、上記の外部情報処理装置から入手したソフトウェア部品の組み上げと実行に必要な情報に基づき、上記情報記憶媒体中に存在しないソフトウェア部品を、上記情報記憶媒体に記憶されたソフトウェア部品から新たに作成し、

実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行すると共に、上記の新しく組み上げたソフトウェア部品を、上記読み出し、書き込み可能な記憶媒体に記憶し、

10

20

30

40

50

次回実行時には、上記記憶媒体に保存された新しいソフトウェア部品を呼び出すことを特徴とする情報処理装置。

【請求項7】請求項1, 2, 3, 4, 5, 又は6記載の情報処理装置であって、アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、

アプリケーションソフトウェアの実行の際に、情報記憶媒体に記憶された上記ソフトウェア部品のうち、機能を実現するのに必要となるソフトウェア部品を選択して組み上げ実行するのに必要な情報を、上記通信装置経由で外部の情報処理装置から入手し、

実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行する処理を、上記情報入力装置による、唯一のアプリケーションソフトウェア実行命令の入力で行なうユーザインタフェースを持つことを特徴とする情報処理装置。

【請求項8】請求項7記載の情報処理装置であって、かつ、外部の情報処理装置から入手するソフトウェア部品の結合情報が、アプリケーションソフトウェアのソースコードであるような情報処理装置であって、

アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、アプリケーションソフトウェアの実行の際に、情報記憶媒体に記憶された上記ソフトウェア部品のうち、

機能を実現するのに必要となるソフトウェア部品を選択して組み上げ実行するのに必要な情報を、上記通信装置経由で外部の情報処理装置から入手し、

実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行する処理を、上記情報入力装置による、唯一のアプリケーションソフトウェア実行命令の入力で行なうユーザインタフェースを持つことを特徴とする情報処理装置。

【請求項9】請求項1, 2, 3, 4, 5, 6, 7, 又は8記載の情報処理装置であって、

かつ、外部の情報処理を2台以上接続し、

アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、アプリケーションソフトウェアの実行の際に、情報記憶媒体に記憶された上記ソフトウェア部品のうち、

機能を実現するのに必要となるソフトウェア部品を選択して組み上げ実行するのに必要な情報を、上記通信装置経由で適切な外部の情報処理装置から入手し、

実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行することを特徴とする情報処理装置。

【請求項10】請求項1, 2, 3, 4, 5, 6, 7,

8, 又は9記載の情報処理装置であって、

情報記憶媒体に、保管しているソフトウェア部品の管理情報を持つことを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理装置に関し、特に、そのアプリケーションプログラムの配布、保管技術に関するものである。

【0002】

【従来の技術】従来の、ネットワーク経由でアプリケーションプログラムを入手し実行する情報処理装置としては、たとえば、Javaアプレットを実行する情報処理装置が知られている。この情報処理装置によれば、アプリケーションのプログラムコードを情報処理装置の記憶媒体に保管することはない。アプリケーションを実行する時には、ネットワーク経由で他の情報処理装置の記憶媒体から情報処理装置にアプリケーションの実行コードを転送してから、その実行コードを実行する。又、必要なアプリケーションの実行が完了した後は、情報処理装置の記憶媒体を圧迫しない目的で、転送された実行コードを破棄する。以上の手順で、Javaアプレットを実行する情報処理装置は、ネットワーク経由でアプリケーションプログラムを入手し、実行する。

【0003】

【発明が解決しようとする課題】さて、前述したJavaアプレットを実行する装置によれば、ネットワーク経由で大規模なアプリケーションソフトウェアを実用的な速度で入手するためには、大容量の回線が必要となる。又、入手したアプリケーションソフトウェアは実行した後、廃棄されてしまうため、アプリケーションプログラムの実行には、その度毎にネットワーク経由でのソフトウェア全体の入手が必要となる。よって、サーバ、クライアントを結ぶ回線の帯域が十分に確保できない場合には、アプリケーションソフトウェアの入手、実行迄に毎回、多くの時間を要することが考えられる。

【0004】そこで、本発明は、帯域幅の十分でないネットワーク経由でもアプリケーションプログラムを高速に入手、実行可能な情報処理装置を提供することを目的とする。

【0005】

【課題を解決するための手段】前記目的達成のために本発明は、演算処理等を行なう中央演算装置と、演算データ、演算結果や制御プログラムを記憶するメモリと、表示データの再生処理を行なう表示装置と、情報記憶媒体及びその情報記憶媒体の駆動装置と、外部の情報処理装置との通信を行なう通信装置を備える情報処理装置において、アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、あるアプリケーションソフトウェアの実行の際に、その機能を実現するのに必要である、情報記憶媒体に記憶されたソフトウェア部品の結合情報を上記通信装置経由で外部の情報処理装置から入手し、実行可能な形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを

実行することを特徴とする情報処理装置を提供する。

【0006】

【発明の実施の形態】以下、本発明に係る情報処理装置の実施例について、図1から図3を用いて説明する。

【0007】図1は、本実施例に係る情報処理装置のハードウェア構成の簡単なブロック図である。

【0008】図2は、本実施例に係る情報処理装置の使用ソフトウェアの簡単な構成図である。

【0009】図3は、本実施例に係る情報処理装置におけるソフトウェア実行時の簡単なフローチャートである。

【0010】図1を用いて本実施例のハードウェア構成を簡単に説明する。

【0011】図1において、101は、情報処理装置であり、各種装置102から113で構成される。102は、情報処理装置の中央演算装置(以下、CPUとする)であり、103は、制御命令及びデータを伝送するシステムバスである。CPU102は、システムバス103から入力された信号に対応した処理を行ない、その処理結果を再びシステムバス103に出力する。

【0012】104は、制御命令及びデータを記憶するランダムアクセスメモリ(RAM)又はリードオンリーメモリ(ROM)である(以下、メモリとする)。105は、データ、制御命令等を記憶したランダムアクセス可能な、例えば、ハードディスク、CD-ROMなどの記憶媒体と、その記憶媒体駆動装置である(以下、記憶媒体と記憶媒体駆動装置を総称して記憶媒体駆動装置とする)。106は、ユーザが命令、データなどの情報を情報処理装置101に対して与えるための、例えばキーボード又はマウスなどの情報入力装置、107は、画像情報 20 を出力するCRT、LCDなどの表示装置、108は、外部情報処理装置121との間で、データ及び制御信号のやり取りをするための通信装置である。109から113は、各装置間を接続するデータ、制御信号用のバスである。

【0013】情報処理装置101は、データ、制御信号用の通信経路114により、外部情報処理装置121に接続される。外部情報処理装置121は、各種装置122から133で構成される。装置122から133は、それぞれ情報処理装置101の装置102から113と 40 同様の機能を持つ装置である。

【0014】尚、図中で各種装置102から113、122から133は、おのおの一つずつ記述されているが、各情報処理装置上に一つと限る訳ではなく、それぞれを複数有するような構成でもよい。

【0015】次に、図2を用いて本実施例の情報処理装置が使用するソフトウェア構成を簡単に説明する。基本となる概念は、ある機能を持つアプリケーションソフトウェアを実現するに当たり、必要となる要素機能をソフトウェア部品として保持することにある。

【0016】本実施例の情報処理装置が使用するソフトウェアは、モジュール構造をとっており、多くのモジュールプログラム(以下、部品プログラムとする)と、そのモジュールプログラムを組み合わせるための情報からなるプログラム(以下、コンテナプログラムとする)からなる。ここで言うソフトウェアのモジュール構造とは、各部品プログラム間のインタフェースが統一されており、容易に入れ替えて使用可能であることを指す。

【0017】部品プログラムの実装方式は特に限定せず、情報処理装置101のCPU102で即時解釈実行可能なマシン語の実行コード(以下、ネイティブコードとする)であってもよいし、情報処理装置101のCPUアーキテクチャに依存しない様なマシン語の実行コード(以下、非依存コードとする)であってもよいし、何らかの高級言語で記述されたプログラムのソースコード(以下、ソースコードとする)であってもよい。又、このうちのどれか一種類でもよいし、複数の組み合わせでもよい。

【0018】尚、生成されるアプリケーションプログラムの汎用性を高くするには、低レベルで単機能な部品プログラムを大量に保持する方法が有利であるので、高級言語の関数レベルのソースコードを部品プログラムとして扱ってもよい。

【0019】コンテナプログラムの実装方法も特に限定せず、なんらかの高級言語記述のプログラムソースコードであってもよいし、マシン語の実行コードであってもよい。又、コンテナプログラムによる部品プログラムの呼び出し方法についても特に限定しない。

【0020】アプリケーションソフトが必要とする機能を実際に実現するためのコードは、部品プログラムに記述されており、コンテナプログラムには実装されていない。コンテナプログラムは、各機能を実現する部品プログラムのローダとして機能する。

【0021】必要な部品プログラムが呼び出されて情報処理装置101のメモリ104にロードされるのは、アプリケーションソフトの実行開始時であってもよいし、部品プログラムの持つ機能が必要となる時点であってもよい。

【0022】部品プログラム、又はコンテナプログラムが、非依存コード、又はソースコードである場合には、ネイティブコードに変換(以下、コンパイルとする)された後に実行される。すなわち、非依存コード、ソースコードの部品プログラムは、各々コンパイルされてネイティブコードとなり、コンテナプログラムの組みあわせ情報に基づき、元々のネイティブコードと共に、実行される。

【0023】尚、非依存コード、ソースコードのコンパイルに必要なソフトウェア(例えば、コンパイラ、ライブラリなど)、及びネイティブコードによる部品プログラムは、情報処理装置101の記憶媒体駆動装置105 50

内にあらかじめ保管してある。又、コンテナプログラムは、外部情報処理装置121の記憶媒体駆動装置125内にあらかじめ保管してある。

【0024】次に、図3を用いて本実施例に係る情報処理装置のソフトウェア実行時の手順を簡単に説明する。

【0025】情報処理装置101のユーザは、情報入力装置106を利用して、あるプログラム(プログラムAとする)の実行開始を入力する。入力された実行開始命令は、バス109、103を経由してCPU102に伝わる。CPU102は、命令を解釈し、バス103、113、通信装置108、通信経路114経由で外部情報処理装置121に、プログラムAのコンテナプログラムを要求する。プログラムAのコンテナプログラムは記憶媒体駆動装置125に保管されており、情報処理装置101の要求に基づき、CPU122は、バス131、123、通信装置128、通信経路114経由で、情報処理装置101に、プログラムAのコンテナプログラムを転送する。CPU102は、プログラムAのコンテナプログラムを通信装置108、バス113、103、112経由でメモリ104に保管した後、そのプログラム情報を解釈する。CPU102は、コンテナプログラムの情報に基づき、記憶媒体駆動装置105から必要な部品プログラムをバス111、103、112経由でメモリ104にロードする。

【0026】尚、コンテナプログラムが実行コードである時、プログラムの実行とはメモリ104にロードされたコンテナプログラムが実行されるということである。コンテナプログラムがなんらかの高級言語記述のプログラムソースコードである時、プログラムの実行とは、ソースコードがなんらかのコンパイラ又はインタープリタなどによって解釈実行されるということである。

【0027】又、前述したように、部品プログラムには、ソースコード、非依存コード、ネイティブコードなどの種類がある。コンテナプログラムの記述に従い、各々の種類に応じてコンパイルなどの処理を行なうことで情報処理装置101上で最終的に、プログラムA全体の実行コードに変換され実行される。

【0028】尚、部品プログラムがソースコードであった場合、コンパイルによって生成されたモジュールをネイティブコードの部品プログラムとして記憶媒体駆動装置105に保管することで、同一プログラムの複数回実行時の実行開始までの時間を短縮することができる。

【0029】以上の手順により、プログラムAが、外部情報処理装置121の記憶媒体駆動装置125に保管されたコンテナプログラムと、記憶媒体駆動装置105に保管された部品プログラムとを用いて、情報処理装置101上で実行される。

【0030】すなわち、本実施例の情報処理装置101におけるプログラムの実行とは、外部情報処理装置121からコンテナプログラムを入手し、記憶媒体駆動装置

105に保管された各部品プログラムを組み合わせた後に実行するということである。ここで、アプリケーションソフトの機能は、部品プログラムの組み合わせで実現できる機能によって規定される。

【0031】本実施例の情報処理装置の特徴を一言で言うと、大量の部品プログラムを情報処理装置101にローカルに保持し、小容量のコンテナプログラムをネットワーク経由で外部情報処理装置121から入手し、情報処理装置101ローカルで実行可能形式に組み上げた後に、目的のプログラムを実行する情報処理装置である。

【0032】尚、本実施例における外部情報処理装置121をネットワークサーバとし、そのネットワークサーバに、複数の情報処理装置101をネットワーククライアントとして接続することで、アプリケーションソフトウェアの配布を容易とするクライアントサーバ・システムを構築しても良い。この場合、コンテナプログラムの情報を更新するだけで、全てのクライアントが使用するアプリケーションを同時に更新することが可能であり、統一的なシステムの維持管理が容易となる利点がある。

【0033】又、あるアプリケーションソフトウェアを情報処理装置101が必要とした際に、外部情報処理装置121、情報処理装置101間の通信経路114に生じるのは、アプリケーションソフトウェアの各種機能を実現するための実際のコードを含まない小容量のコンテナプログラムだけの転送なので、通信経路114に大きな負荷を掛けることがない。これは、外部情報処理装置121に、複数の情報処理装置101を接続するクライアントサーバ・システムとして運用する際に大きな利点となる。又、ローカルユーザにとってもプログラムの入手に掛かる時間が小さいことは作業効率の点で有益である。

【0034】又、情報処理装置101のユーザは、アプリケーションの実行時に、そのプログラムのシステム上の実際のファイリングを意識する必要がなく、運用が容易である。

【0035】本発明の情報処理装置101では、似通った機能を持つソフトウェアで記憶媒体駆動装置105の容量が不要に圧迫されることがない。すなわち、ある機能は特定の部品プログラムに集約される。又、アプリケーションソフトウェアの機能を実際に実現するコードである部品プログラムは、使い捨てにされることがなく、複数のアプリケーションプログラムから利用されるので、記憶媒体駆動装置105の使用効率が良い。

【0036】前述したように、情報処理装置101は部品プログラムを記憶媒体駆動装置105に保管するが、この部品プログラムを保管するための記憶媒体駆動装置105を例えばCD-ROMドライブのような読み出し専用媒体とし、部品プログラムをCD-ROM媒体の形式によって配布してもよい。

【0037】本実施例の情報処理装置でのアプリケーシ

ソフトウェアの更新とは、コンテナプログラムの更新と、部品プログラムの更新に分けられる。このうち、コンテナプログラムの更新は、外部情報処理装置121の記憶媒体駆動装置125上のコンテナプログラムを更新し、通信経路114を経由して情報処理装置101に配信することで行うことができる。又、部品プログラムの更新は、前記、CD-ROMドライブのような可換記憶媒体を用いた配布により行なうことができる。

【0038】以下、本発明に係る情報処理装置の第2の実施例について、図1、図2及び図4を用いて説明する。

【0039】図4は、本実施例に係る情報処理装置におけるソフトウェア実行時の簡単なフローチャートである。

【0040】本実施例に係る情報処理装置の構成は、前記した本発明の第1の実施例に係る情報処理装置と基本的には同様であるが、図1における記憶媒体駆動装置105が例えばハードディスクドライブのような読み書き可能な媒体であることと、外部情報処理装置121の記憶媒体駆動装置125に、コンテナプログラムに加えて、部品プログラムも保管していることの2点が異なる。本実施例の情報処理装置の使用するソフトウェアは、図2の構成と同様である。

【0041】情報処理装置101のユーザは、情報入力装置106を利用して、あるプログラム(プログラムBとする)の実行開始を入力する。入力された実行開始命令は、バス109、103を経由してCPU102に伝わる。CPU102は、命令を解釈し、バス103、111を経由して、プログラムBのコンテナプログラムが記憶媒体駆動装置105上に存在するかどうかを確認する。

【0042】目的のコンテナプログラムが同記憶媒体駆動装置上に存在する場合には、バス111、103、112経由でメモリ104にコンテナプログラムをロードし、解釈する。

【0043】目的のコンテナプログラムが同記憶媒体駆動装置上に存在しない場合には、バス103、113、通信装置108、通信経路114経由で外部情報処理装置121に、プログラムBのコンテナプログラムを要求する。プログラムBのコンテナプログラムは記憶媒体駆動装置125に保管されており、情報処理装置101の要求に基づき、CPU122は、バス131、123、通信装置128、通信経路114経由で、情報処理装置101に、プログラムBのコンテナプログラムを転送する。CPU102は、プログラムBのコンテナプログラムを通信装置108、バス113、103、バス111経由で記憶媒体駆動装置105上に、コンテナプログラムを保管する。又、バス112経由でメモリ104にコンテナプログラムをロードし、解釈する。

【0044】CPU102は、コンテナプログラムの情

報に基づき、必要な部品プログラムが記憶媒体駆動装置105上に存在するかどうかを確認する。

【0045】目的の部品プログラムが同記憶媒体駆動装置上に存在する場合には、記憶媒体駆動装置105から必要な部品プログラムをバス111、103、112経由でメモリ104にロードする。

【0046】目的の部品プログラムが同記憶媒体駆動装置上に存在しない場合には、前記したコンテナプログラムの入手方法と同様の手順で、外部情報処理装置121から、部品プログラムを入手し、記憶媒体駆動装置105上に保管し、かつメモリ104にロードする。

【0047】次に、実施例1と同様の手順で、コンテナプログラムの記述に従い、各々の種類に応じてコンパイルなどの処理を行なうことで情報処理装置101上で、最終的な実行コードに変換され実行される。

【0048】記憶媒体駆動装置105を読み書き可能な記憶媒体とすることで、通信経路114経由で外部情報処理装置121から入手したコンテナプログラムを、情報処理装置101の記憶媒体駆動装置105上に保管することが可能となり、同一プログラムの複数回実行時の実行開始までの時間を短縮することができる。

【0049】尚、本発明の第1の実施例と同様にして、情報処理装置101、外部情報処理装置121によるクライアントサーバシステムを構築している場合に、あるアプリケーションの作成時に必要な機能を実現する部品プログラムの追加、更新といった管理がサーバ上のみで実行できるので、統一的なシステムの維持管理が容易となる利点がある。

【0050】本発明の第1の実施例と同様に、本実施例の情報処理装置でのアプリケーションソフトウェアの更新も、コンテナプログラムの更新と、部品プログラムの更新に分けられる。このうち、コンテナプログラムの更新は、第1の実施例の手順と同様にして実現できる。又、部品プログラムの更新は、第1の実施例と同様にCD-ROMドライブのような可換記憶媒体を用いた配布により行なうことができると共に、外部情報処理装置121の記憶媒体駆動装置125上に更新した部品プログラムを保管し、通信経路114を経由して情報処理装置101の記憶媒体駆動装置105上に保管することでも実現できる。

【0051】以下、本発明に係る情報処理装置の第3の実施例について説明する。

【0052】以下、本発明の第3の実施例について、図1、図2及び図5を用いて説明する。

【0053】本実施例に係る情報処理装置の構成は、前記した本発明の第1の実施例に係る情報処理装置と、基本的に同様であり、使用するソフトウェアも、図2の構成と同様である。ただし、本実施例の扱うコンテナプログラムは、なんらかの高級言語記述のプログラムソースコードである。

【0054】図5は、本実施例に係る情報処理装置の表示画面の一例である。

【0055】例えば、情報処理装置101上で実行可能なプログラムの一覧を表示装置107の画面上に表示する。ユーザは、例えば、マウスである情報入力装置106を用い、画面上に表示された、あるプログラム(プログラムCとする)の実行開始を入力する。この入力に対し、以下、第2の実施例と同様の手順により、外部情報処理装置121から情報処理装置101へプログラムCの実行に必要なコンテナプログラムが転送される。

【0056】このコンテナプログラムは前述したとおり、プログラムソースコードであり実行属性ではない。情報処理装置101は、コンテナプログラムの入手後その記述に従い、記憶媒体駆動装置上に保管された部品プログラムを読み込み実行コードを生成し、即実行する。

【0057】すなわち、ユーザは情報処理装置101に対してあるプログラムの実行を命令することだけで、情報処理装置101が自動的に、通信経路114を経由して外部情報処理装置121からのプログラムソースコードの入手からコンパイル、実行まで行なう。ユーザがその動作手順を意識する必要はなく、ネットワークを介してのプログラムの実行が容易である。

【0058】尚、例えばHTMLなどのスクリプト記述によって、情報処理装置上で実行可能なプログラムの一覧表示を行なう場合には、スクリプト中に通信経路を経由して入手するプログラムソースコードをコンパイルし、生成された実行ファイルの実行を行なう"命令"を記述することでもこの機能は実現できる。

【0059】以上のように本第3実施例によれば、プログラムソースコードが情報処理装置上にはなく、通信経路を経由して入手すること、又、入手したコンテナプログラムが実行コードではなく、情報処理装置上で部品プログラムを用いたコンパイルの操作が行なわれることをユーザが意識せずに済むインタフェースを提供することが可能になる。

【0060】尚、本発明の情報処理装置では、コンテナプログラムが呼び出す部品プログラムの組み合わせを変更することで、多種類のアプリケーションソフトウェアを実現可能である。あるアプリケーションに必要な部品プログラムを高速に呼び出すには、記憶媒体駆動装置105上に保管される部品プログラムの管理が重要である。記憶媒体駆動装置105上で部品プログラムが保管されている位置、記憶媒体駆動装置105上にどの種類の部品プログラムが保管されているか、部品ファイルに更新があった場合のバージョンの管理などを行なうデータベースを情報処理装置101の記憶媒体駆動装置105に設け、コンテナプログラムが要求する部品プログラムの情報と比較することで、上記の部品プログラムの管理が容易となる。

【0061】本発明の情報処理装置では、部品プログラ

ムを保管する記憶媒体が比較的低速(例えばCD-ROMなど)であっても、全ての実行コードを含んだアプリケーションプログラム全体をネットワーク経由で入手するよりは高速に、実行コードの入手が可能である。特に本発明の情報処理装置の効果は、通信経路が低速であるほど顕著に現れる。すなわち、本発明の情報処理装置が有効である条件は、あるアプリケーションソフトウェアを実行するのに必要なコンテナプログラムの通信経路での入手に要する時間と、部品プログラムのコンパイルに要する時間の合計が、同アプリケーションソフトウェアの実行コード全体を通信経路で入手するのに要する時間よりも短いことである。

【0062】尚、本発明の情報処理装置が使用する、コンテナプログラムの情報に基づく部品プログラムのコンパイルと組み合わせ処理の機能を、情報処理装置の使用するOSに組み込んでもよい。部品プログラムをライブラリとして記憶媒体駆動装置上に標準装備し、コンテナプログラムを外部情報処理装置から入手する。コンテナプログラムは、前述したとおり各機能の実行コードを含まない形で配布されるので、アプリケーションプログラムの小容量でのパッケージが可能である。これにより、OSは多数の部品プログラムと、その実行環境を標準装備し、アプリケーションは共通の部品プログラムの組み合わせで成立する。

【0063】OSに上記機能を組み込むことで、どの情報処理装置でも間違いなくアプリケーションソフトウェアが実行可能であることが保証されるので、クライアントサーバ・システムとして運用する際の利便性が高まる効果がある。

【0064】

【発明の効果】以上、説明してきたように本発明によれば、アプリケーションソフトウェアの部品となるプログラムを情報記憶媒体にあらかじめ記憶しておき、あるアプリケーションソフトウェアの実行の際に、その機能を実現するのに必要である、情報記憶媒体に記憶されたソフトウェア部品の結合情報を通信装置経由で外部の情報処理装置から入手し、実行可能形式に組み上げた後に、上記目的となるアプリケーションソフトウェアを実行することを特徴とする情報処理装置を提供することができる。

【図面の簡単な説明】

【図1】本発明の第1実施例に係る情報処理装置のハードウェア構成の簡単なブロック図である。

【図2】本発明の第1実施例に係る情報処理装置の使用ソフトウェアの簡単な構成図である。

【図3】本発明の第1実施例に係る情報処理装置におけるソフトウェア実行時の簡単なフローチャートである。

【図4】本発明の第2実施例に係る情報処理装置におけるソフトウェア実行時の簡単なフローチャートである。

【図5】本発明の第3実施例に係る情報処理装置の表示

画面の一例である。

【符号の説明】

101…情報処理装置、102…中央演算装置、103…システムバス、104…メモリ、105…記憶媒体駆動装置、106…情報入力装置、107…表示装置、108…通信装置、109…信号バス、110…信号バス、111…信号バス、112…信号バス、113…信号バス、114…通信経路、121…外部情報処理装置、122…中央演算装置、123…システムバス、124…メモリ、125…記憶媒体駆動装置、126…情報入力装置、127…表示装置、128…通信装置、129…信号バス、130…信号バス、131…信号バス、132…信号バス、133…信号バス。

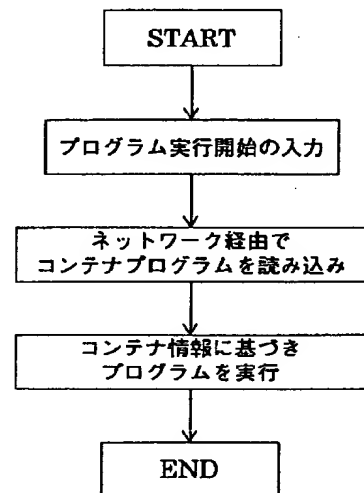
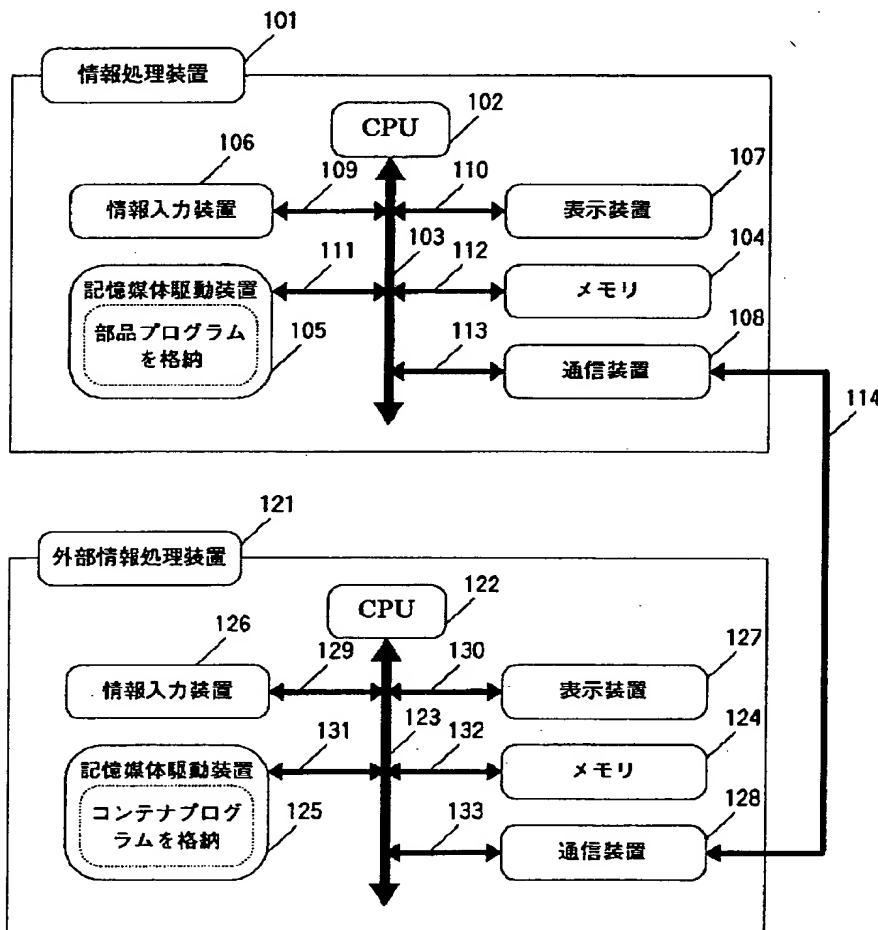
13…信号バス、114…通信経路、121…外部情報処理装置、122…中央演算装置、123…システムバス、124…メモリ、125…記憶媒体駆動装置、126…情報入力装置、127…表示装置、128…通信装置、129…信号バス、130…信号バス、131…信号バス、132…信号バス、133…信号バス。

【図1】

【図3】

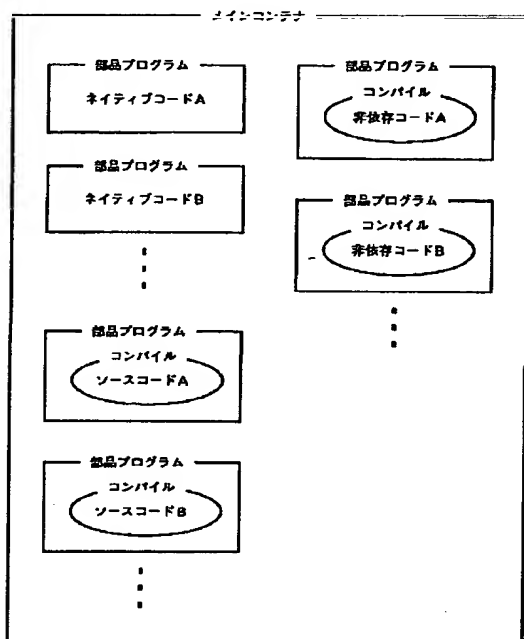
図 1

図 3



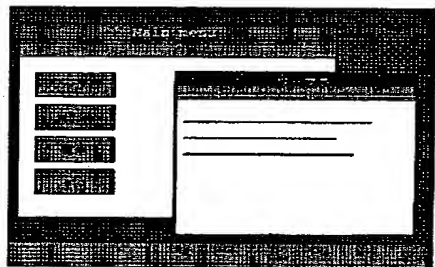
【図2】

図 2



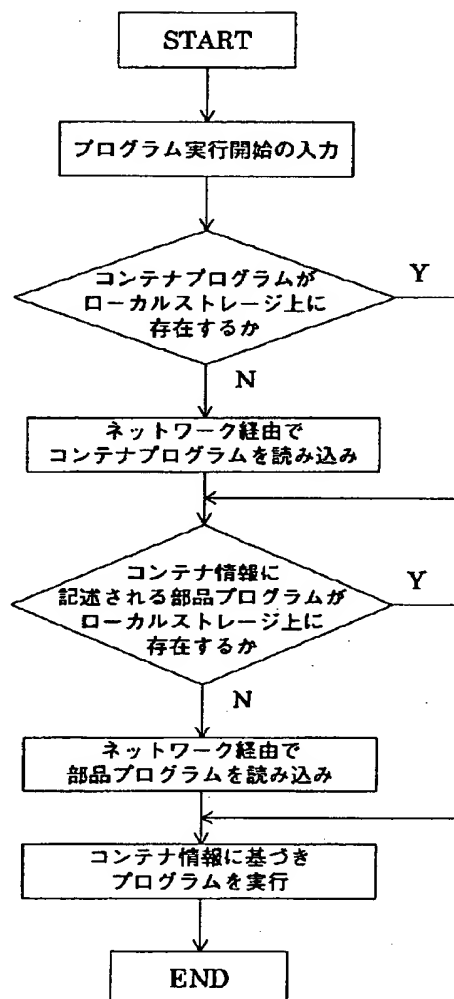
【図5】

図 5



【図4】

図 4



フロントページの続き

- (72)発明者 大和田 徹
 神奈川県川崎市麻生区王禅寺1099番地株式
 会社日立製作所システム開発研究所内
- (72)発明者 滝田 功
 神奈川県川崎市麻生区王禅寺1099番地株式
 会社日立製作所システム開発研究所内
- (72)発明者 池田 牧子
 神奈川県川崎市麻生区王禅寺1099番地株式
 会社日立製作所システム開発研究所内

- (72)発明者 永井 靖
 神奈川県川崎市麻生区王禅寺1099番地株式
 会社日立製作所システム開発研究所内
- (72)発明者 小沼 智
 神奈川県横浜市戸塚区吉田町292番地株式
 会社日立画像情報システム内
- (72)発明者 高木 徹夫
 神奈川県横浜市戸塚区吉田町292番地株式
 会社日立マイクロソフトウェアシステムズ
 内